Final project report

1..why this topic:

The topic of my project is automated essay grading.Automated essay grading is crucial for its scalability, efficiency, and consistency, enabling the rapid and unbiased evaluation of large volumes of essays. It reduces costs and provides immediate feedback, helping students learn faster. By eliminating human biases and offering data-driven insights, it supports educators in tailoring their teaching strategies. Additionally, it allows teachers to focus on higher-order tasks and enhances educational tools, ensuring standardized and fair assessments.The dataset is Essays with human graded scores.

2.the steps of doing this project:

1)First I tried to treat it as a classification problem:

I.Neural Network method

Steps:

Preprocessing:Convert text to lowercase, remove non-alphabetic characters, tokenize the text, remove stopwords, and apply Porter stemming. Returns a string and a list of tokens.

Training: train a word2vec model with a vector size of 100 on the tokenized document

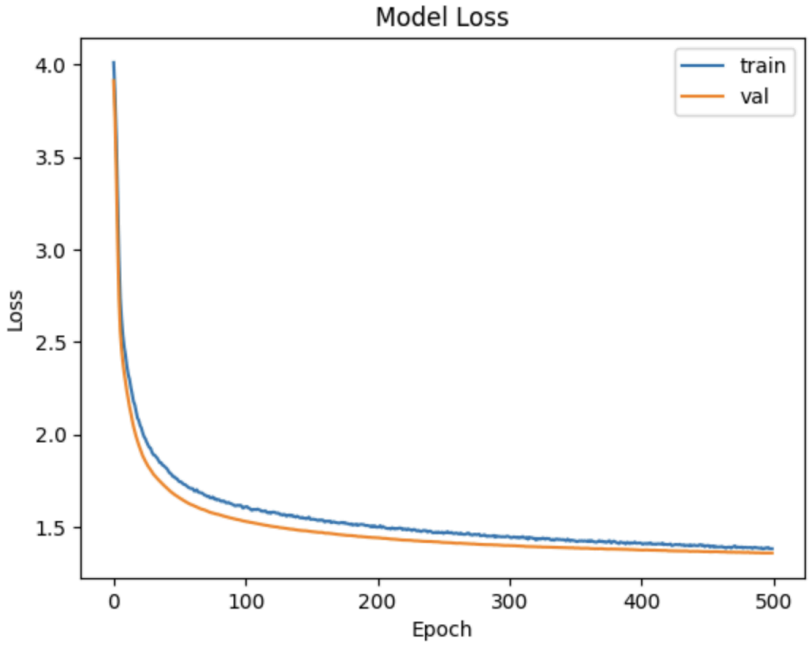
Training-testing split:split the data into training 0.75 and testing 0.25.

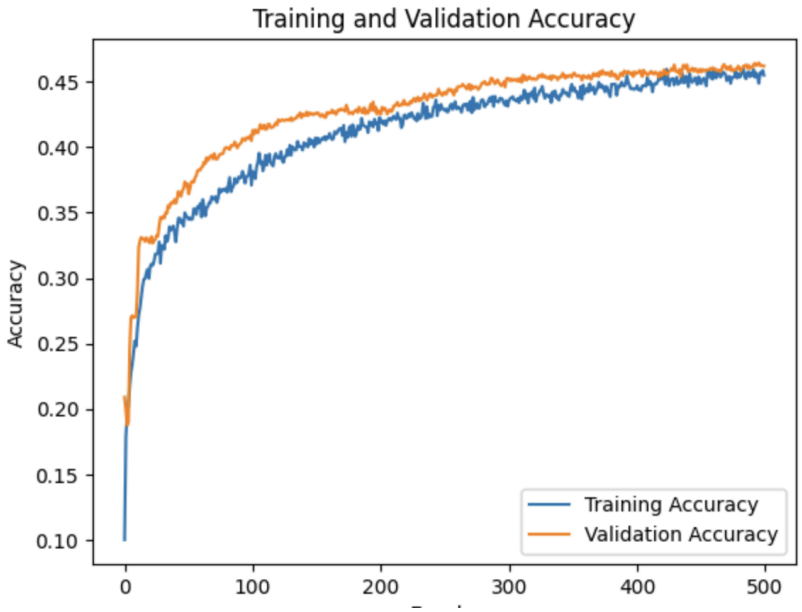
Turning the documents into vectors and using a one-hot encoding ‘essay_set’ feature then combines these two.

Model: a sequential neural network with one input layer and two hidden layers with dropout and one output layer with softmax activation function.Using the Adam optimizer and sparse categorical cross-entropy loss and setting the learning rate to 0.0001

Training the model for 500 epochs with early stopping to prevent overfitting

Result: the accuracy is only 46.2%,the model loss and the training,validation accuracy are in the pictures below:





II. LSTM method:

Steps:

Preprocessing: similar like previous one but convert both essay and essay_set into sequences of words using word2vec and pad the sequence to a max length for consistency,using separate embeddings layers for essay and essay_set sequences

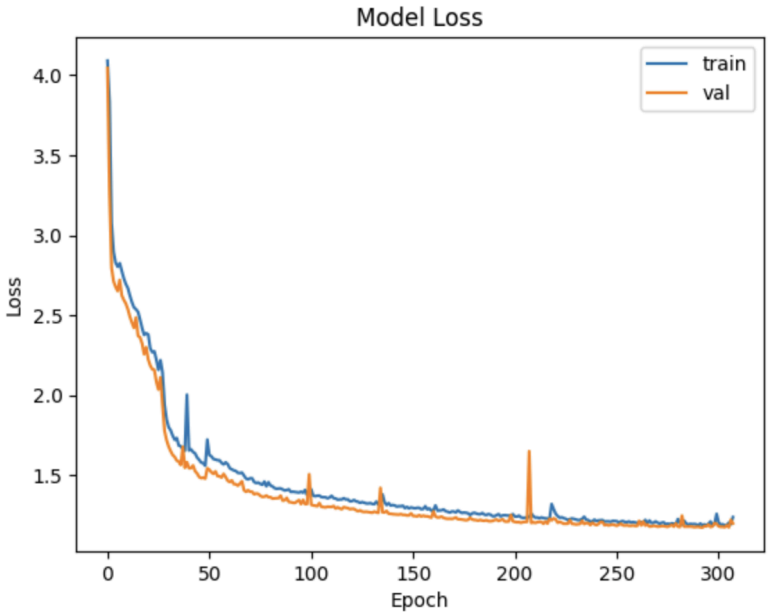
Training:use LSTM layers to capture the sequential nature of the text,employ embedding layers initialized with pre-trained word2vec embeddings,use two separate inputs processed through LSTM.

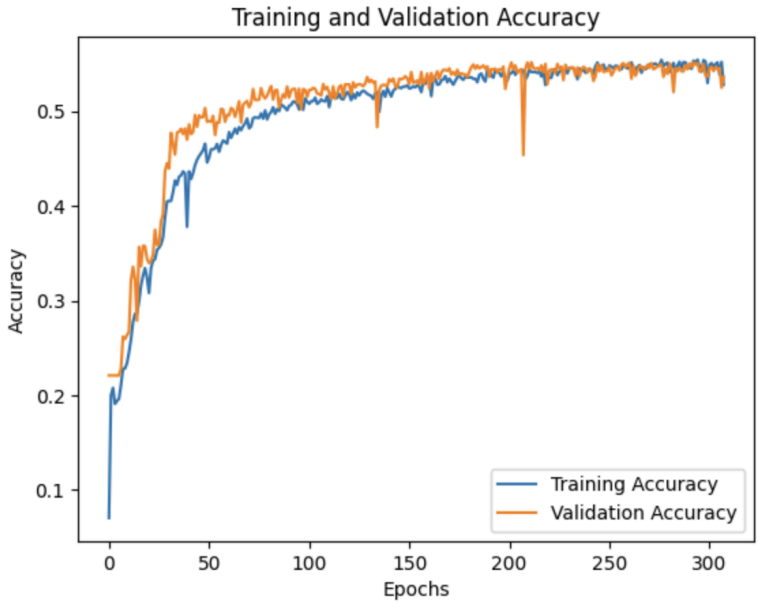
Training-testing split:split the data into training 0.75 and testing 0.25.

Model: a LSTM model with two input layers, an embedding layer using the pre_trained embedding matrix,2 LSTM layers for both inputs, global max pooling layers to reduce the dimensionality then concatenate the pooled output and using a dense layer before dropout and output layer.

Training the model for 500 epochs with early stopping to prevent overfitting

Result:the accuracy is 55.2%,the model loss and the training,validation accuracy are in the pictures below:





2)Then I tried to treat it as a regression problem:

I.First I tried some machine learning methods:

The training-testing split of them are all the same as the previous:0.75 for training and 0.25 for testing.

Data preprocessing: using “essay_id”,”essay_set”,”essay”,”domain1_score” to create “asap”dataframe;using “minmax_scaler” to scale scores to [0,1] range and use the min and max for each essay set;using “inverse_scaler” convert scales to original range.

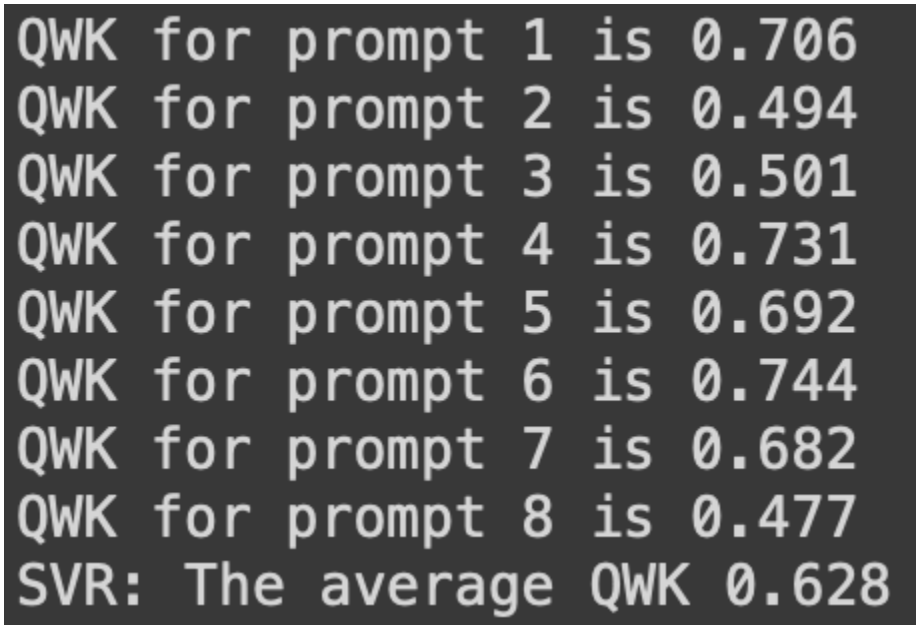
Using “scale_dataset” to scale essay scores in “minmax_scaler” and add a new “nscore” column.

Using TF-IDF to vectorization the dataset

Method 1: Support Vector Regression(SVR)

Calculate qwk as the cohen_kappa_score using weight='quadratic', using the average qwk score as the final score

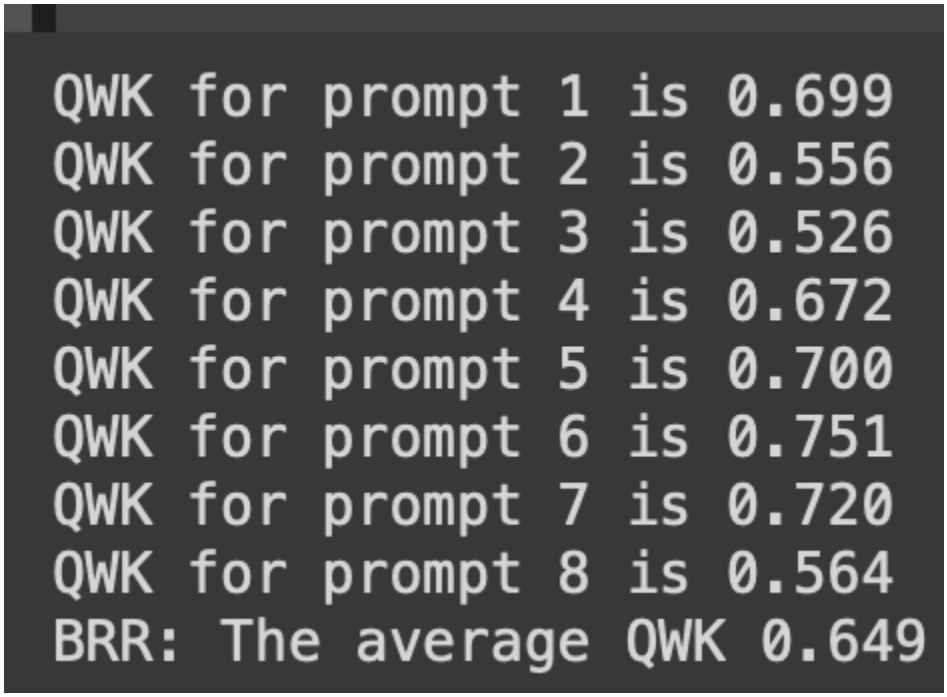
Qwk for SVR is 0.628



Method 2: Bayesian Ridge Regression(BRR):

Using the same qwk score

Qwk for BRR is 0.649



Method 3: XGBoost(XGB)

Qwk for XGB is 0.689

```
QWK for prompt 1 is 0.771
QWK for prompt 2 is 0.613
QWK for prompt 3 is 0.610
QWK for prompt 4 is 0.746
QWK for prompt 5 is 0.712
QWK for prompt 6 is 0.769
QWK for prompt 7 is 0.730
QWK for prompt 8 is 0.560
XGB: The average QWK 0.689
```

II. Then I tried to add lexical features to improve the prediction accuracy:
The lexical features I used are character-based,punctuation-based,word-based,sentence-based,number-based,correct words and part of speech feathers.I also used a dataframe called ‘asap’ to extract the defined features and store them.
I added this lexical feature into the previous 3 methods and saw an improvement about the qwk result
SVR: the qwk is 0.716

```
 Prompt   Val    Test
  1       0.812  0.812
  2       0.672  0.672
  3       0.533  0.533
  4       0.755  0.755
  5       0.742  0.742
  6       0.778  0.778
  7       0.731  0.731
  8       0.704  0.704
SVR: The average QWK 0.716
```

BRR: the qwk is 0.754

Prompt	Val	Test
1	0.817	0.817
2	0.686	0.686
3	0.645	0.645
4	0.770	0.770
5	0.784	0.784
6	0.814	0.814
7	0.777	0.777
8	0.741	0.741

BRR: The average QWK 0.754

XGB: the qwk is 0.751

Prompt	Val	Test
1	0.815	0.815
2	0.687	0.687
3	0.673	0.673
4	0.748	0.748
5	0.802	0.802
6	0.808	0.808
7	0.772	0.772
8	0.700	0.700

XGB: The average QWK 0.751

3.summary：

In this project, I have used classification and regression methods separately for automated essay grading problem. In all the methods, I divided the initial dataset into

0.75 for training and 0.25 for testing. For classification, I tried sequential neural network and LSTM approach for 500 iterations with accuracy of 46.21% and 55.15% respectively. For regression, I first tried using SVR, BRR and XGB methods and calculated their quadratic weighted kappa coefficients (qwk) as 0.628, 0.649, 0.689 respectively. Based on this, I tried to add lexical features to improve the qwk values, and after applying the new features the qwk values were After applying the new features, the qwk values are 0.716, 0.754, and 0.751, which are significantly improved compared to the previous data.